МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Специальность ПО-5

Выполнила

А. А. Нерода,

студентка группы ПО-5

Проверил

А. А. Крощенко,

ст. преп. кафедры ИИТ,

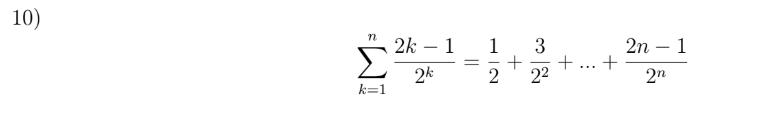
Брест 2022

**Вариант 10**

**Цель работы:** приобрести навыки написания простого оконного многопоточного приложения с использованием

Java API.

**Задание:** Разработать оконное приложение с использованиемJava API,использующее одинвспомогательный поток, вычисляющий заданную сумму и выполняющий вывод результата вычисления (как конечный, так и промежуточные) в любой визуальный компонент. Все исходные данные вводятся в соответствующие визуальные компоненты. В программе должны быть предусмотрены функции приостановки, возобновления и полной остановки выполнения потока с выводом соответствующего сообщения. В случае быстрого выполнения потока и, как следствие, невозможности демонстрации функций приостановки, продумать искусственное «торможение» потока для достижения заданных целей. Обработать исключения.



Код программы:

package com.company;  
  
import javafx.application.Application;  
import javafx.application.Platform;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Alert;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
import javafx.scene.layout.GridPane;  
import javafx.scene.layout.HBox;  
import javafx.scene.text.Font;  
import javafx.scene.text.FontWeight;  
import javafx.scene.text.Text;  
import javafx.stage.Stage;  
public class Main extends Application {  
 Text sceneTitle = new Text("Calculate function");  
 Label labelForN = new Label("Enter n value:");  
 Label labelForResult = new Label("Value is not given");  
 TextField textFieldForN = new TextField();  
 HBox hbox = new HBox();  
 Button calculateResult = new Button("Calculate result");  
 Button pauseThread = new Button("Pause thread");  
 Button stopThread = new Button("Stop thread");  
 private double sum = 0;  
 private int counter = 1;  
  
 private final Object lock = new Object();  
 private volatile boolean paused = true;  
 private Thread calculator = new Thread(() -> {  
 while (true) {  
 work();  
 }  
 });  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
 @Override  
 public void start(Stage primaryStage) {  
 calculator.start();  
 primaryStage.setTitle("Lab 8");  
 GridPane grid = new GridPane();  
 grid.setAlignment(Pos.*CENTER*);  
 grid.setHgap(10);  
 grid.setVgap(10);  
 grid.setPadding(new Insets(25, 25, 25, 25));  
 sceneTitle.setFont(Font.*font*("Tahoma", FontWeight.*NORMAL*, 20));  
 grid.add(sceneTitle, 0, 0, 2, 1);  
 grid.add(labelForN, 0, 1);  
 grid.add(textFieldForN, 1, 1);  
 grid.add(labelForResult, 1, 2);  
 hbox.setPadding(new Insets(15, 12, 15, 12));  
 hbox.setSpacing(10);  
 hbox.setAlignment(Pos.*BOTTOM\_RIGHT*);  
 hbox.getChildren().addAll(calculateResult, pauseThread, stopThread);  
 grid.add(hbox, 1, 4);  
 Scene scene = new Scene(grid, 800, 300);  
 primaryStage.setScene(scene);  
 primaryStage.show();  
 calculateResult.setOnAction(e -> {  
 startTask();  
 });  
 pauseThread.setOnAction(e -> {  
 paused = !paused;  
 });  
 stopThread.setOnAction(e -> {  
 done();  
 });  
 }  
 public void startTask() {  
 try {  
 counter = Integer.*parseInt*(textFieldForN.getText());  
 paused = !paused;  
  
 synchronized (lock) {  
 lock.notifyAll();  
 }  
 } catch (NumberFormatException e) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle("Alert");  
 alert.setHeaderText("Information Alert");  
 String s = "Incorrect number format";  
 alert.setContentText(s);  
 alert.show();  
 }  
 }  
 private void pauseTask() {  
 synchronized (lock) {  
 while (paused) {  
 try {  
 lock.wait();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
 private void done() {  
 paused = true;  
 this.sum = 0;  
 this.counter = 0;  
 this.textFieldForN.setText("");  
 }  
 private void work() {  
 pauseTask();  
 for (int k = 1; k <= counter; k++) {  
 pauseTask();  
 sum = ((2 \* k - 1) / (Math.*pow*(2, k)));  
 final String status = "Calculating k = " + k + ", sum = " + sum;  
 Platform.*runLater*(() -> labelForResult.setText(status));  
 try {  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 done();  
 }  
}

Результат работы программы:

